(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
20 July 2006 (20.07.2006)

(10) International Publication Number
WO 2006/075895 A1

(51) International Patent Classification:
G06T 9/00 (2006.01)

(21) International Application Number: PCT/KR2006/000151

(22) International Filing Date: 13 January 2006 (13.01.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10-2005-0003832 14 January 2005 (14.01.2005) KR
10-2005-0102222 28 October 2005 (28.10.2005) KR

(71) Applicant *(for all designated States except US)*: ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE [KR/KR]; 161 Gajeong-dong, Yuseong-gu, Daejeon 305-350 (KR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CHANG, Eun Young** [KR/KR]; 1124-10, Jinbuk-dong, Deokjin-gu, Jeonju, Jeollabuk-do 561-160 (KR). **AHN, Chung Hyun** [KR/KR]; Hyundai Apt. 101-705, Doryong-dong, Yuseong-gu, Daejeon 305-340 (KR). **JANG, Euee Seon** [KR/KR]; 17 Haengdang-dong, Seongdong-gu, Seoul 133-791 (KR). **KIM, Mi Ja** [KR/KR]; 17 Haengdang-dong, Seongdong-gu, Seoul 133-791 (KR). **KIM, Dai Yong** [KR/KR]; 17 Haengdang-dong, Seongdong-gu, Seoul 133-791 (KR). **LEE, Sun Young** [KR/KR]; 17 Haengdang-dong, Seongdong-gu, Seoul 133-791 (KR).

(74) Agent: **SHIN, Young Moo**; Ace Tower 4th Floor 1-170, Soonhwa-dong Chung-gu, Seoul 100-130 (KR).

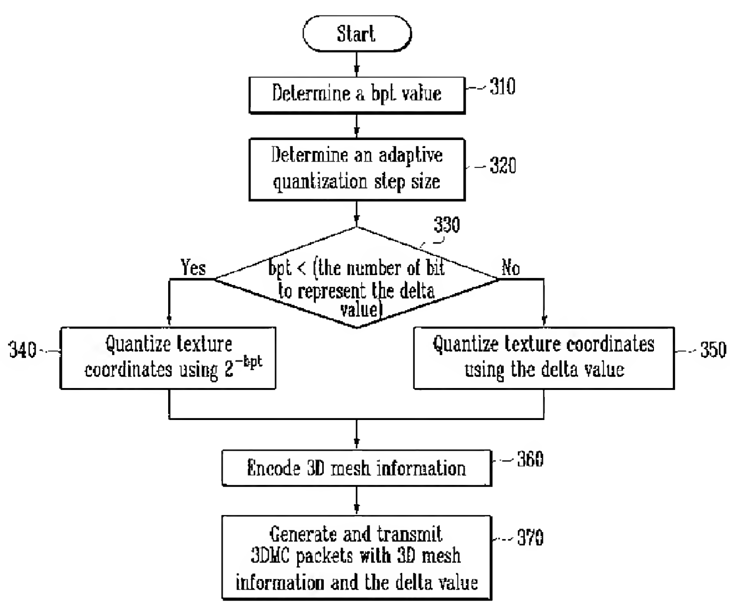
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: METHOD OF ENCODING AND DECODING TEXTURE COORDINATES IN THREE-DIMENSIONAL MESH INFORMATION FOR EFFECTIVE TEXTURE MAPPING



(57) Abstract: Provided is a method of encoding and decoding texture coordinates of 3D mesh information. The method of encoding texture coordinates in 3D mesh information includes the steps of: setting an adaptive quantization step size used for quantizing the texture coordinates; quantizing the texture coordinates using the adaptive quantization step size; and encoding the quantized texture coordinates.

WO 2006/075895 A1

Description

METHOD OF ENCODING AND DECODING TEXTURE COORDINATES IN THREE-DIMENSIONAL MESH INFORMATION FOR EFFECTIVE TEXTURE MAPPING

Technical Field

[1] The present invention relates to a method of encoding and decoding three-dimensional ("3D") mesh information, and more particularly, texture coordinates in the 3D mesh information, which guarantees the lossless compression of them for effective texture mapping.

Background Art

[2] 3D graphics have been widely used, but it has a limitation to its use range due to heavy amount of information. A 3D model is expressed by the mesh information, which includes geometry information, connectivity information, and attribute information having normal, color and texture coordinates. The geometry information is comprised of three coordinate information expressed by floating points, and the connectivity information is expressed by an index list, in which three or more geometric primitives form one polygon. For example, if it is assumed that the geometry information is expressed by the floating points of 32 bits, 96 bits (i.e., 12B) are needed to express one geometry information. That is, 120KB bits are needed to express a 3D model having ten thousand vertices with only geometry information, and 1.2MB are needed to express a three-dimensional model having hundred thousand vertices. The connectivity information requires much memory capacity to store the polygonal 3D mesh, since twice or more duplication is allowed.

[3] For the reason of the huge amount of information, the necessity of compression has been raised. To this end, the 3D mesh coding (3DMC) which is adopted as a standard of International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) in Moving Picture Expert Group-Synthetic and Natural Hybrid Coding (MPEG-4-SNHC) field improves transmission efficiency by encoding/decoding 3D mesh information expressed by IndexFaceSet(IFS) in a Virtual Reality Modeling Language (VRML) file.

[4] FIG. 1 is a conceptual diagram of a typical 3DMC coding. As shown in FIG. 1, IFS data in VRML file is transformed into a 3DMC bit stream through quantization and encoding processes. The 3DMC bitstream is reconstructed by inverse quantization and decoding processes.

[5] As the texture mapping is widely used in 3D games or interactive graphic media, a need of lossless compression for the texture coordinates in IFS is being gradually

increased. However, a conventional 3DMC has a weak not to guarantee the lossless compression of the texture coordinates after decoding through the quantization process in encoding

[6] FIG. 2 is a conceptual diagram illustrating a texture mapping error after encoding and decoding by the conventional 3DMC. FIG. 2 shows a texture mapping error occurring when an integer texture coordinate (400,800) in an original texture image is transformed into a real number between "0" and "1" in the VRML file, is subjected to encoding and decoding processes, and then is reconstructed to a different integer texture coordinate (401,801) during rendering.

[7] As described above, the conventional 3DMC has a problem in that the integer texture coordinate of the original texture image is mapped to the real number and quantized, but is not reconstructed to the original integer texture coordinate in the reconstruction process.

Disclosure of Invention

Technical Problem

[8] The present invention is directed to a method of encoding/decoding texture coordinates, which is capable of allowing the texture coordinate to be losslessly reconstructed for accurate texture mapping.

[9] The present invention is also directed to a method of efficiently encoding/decoding texture coordinates by adaptively adjusting the quantization step size (or delta value) used for the texture coordinate quantization.

Technical Solution

[10] A first aspect of the present invention is to provide a method of encoding texture coordinates in 3D mesh information. The method comprises the steps of: determining an adaptive quantization step size used for texture coordinate quantization; quantizing the texture coordinates using the adaptive quantization step size; and encoding the quantized texture coordinates.

[11] Preferably, the adaptive quantization step size may be determined as the inverse of the texture image size or may be determined using the texture coordinates.

[12] The step of determining the adaptive quantization step size comprises the sub-steps of: checking whether the texture image size information exists or not; determining the inverse of the texture image size as a first quantization step size when the texture image size information exists; obtaining a second quantization step size using the texture coordinates; checking whether the second quantization step size is a multiple of the first quantization step size; determining the second quantization step size as the adaptive quantization step size when it is determined that the second quantization step size is a multiple of the first quantization step size; and determining the first

quantization step size as the adaptive quantization step size when it is determined that the second quantization step size is not a multiple of the first quantization step size.

[13] A second aspect of the present invention is to provide a method of encoding 3D mesh information. The method comprises a first encoding step for encoding a texture coordinate in the 3D mesh information according to above-described encoding method; a second encoding step for encoding remaining information of the 3D mesh information; and a step of producing 3D mesh coding (3DMC) packets which contain the 3D mesh information obtained by the first and second encoding steps and an adaptive quantization step size.

[14] A third aspect of the present invention is to provide a method of decoding texture coordinates in 3DMC packets, which comprise the steps of: extracting adaptive quantization step size information from the 3DMC packet; inverse-quantizing the texture coordinates in the 3DMC packet using the extracted adaptive quantization step size; and decoding the inverse-quantized texture coordinates.

[15] A fourth aspect of the present invention is to provide a 3DMC decoding method, which comprises (i) decoding texture coordinates in 3DMC packets according to above-described decoding method;(ii) decoding the remaining information of the 3DMC packets; and (iii) reconstructing a 3D model based on 3D mesh information generated from the decoding results in the steps (ii) and (iii).

Advantageous Effects

[16] The method of encoding/decoding the 3D mesh information for the effective texture mapping according to the present invention achieves lossless reconstruction of the texture coordinates by adaptively adjusting the quantization step size for quantizing the texture coordinates, thereby guaranteeing the accurate texture mapping.

Brief Description of the Drawings

[17] FIG. 1 is a conceptual diagram of a typical 3DMC coding;

[18] FIG. 2 is a conceptual diagram illustrating a texture mapping error after encoding and decoding by a conventional 3DMC scheme;

[19] FIG. 3 is a flowchart illustrating a 3DMC encoding process with texture coordinate quantization according to an embodiment of the present invention;

[20] FIG. 4 is a flowchart illustrating the process for calculating an adaptive quantization step size (i.e., delta value) for texture coordinate quantization according to an embodiment of the present invention;

[21] FIG. 5 is a flowchart illustrating a 3D decoding process according to an embodiment of the present invention;

[22] FIGS. 6a to 6d show the results of quantizing texture coordinates according to the conventional 3DMC scheme and the present invention;

[23] FIG. 7a illustrates the rendering result after the encoding/decoding is performed according to the conventional 3DMC scheme; FIG. 7b illustrates the rendering result after the encoding/decoding is performed using the inverse of the image size as the delta value according to one embodiment of the present invention; and FIG. 7c illustrates the rendering result after the encoding/decoding is performed using the delta value calculated from the texture coordinates according to another embodiment of the present invention; and

[24] FIG. 8 shows a structure of a 3DMC packet header with a flag "delta_flag" indicating whether there is adaptive quantization step size (delta) information in a 3DMC packet according to an embodiment of the present invention.

Mode for the Invention

[25] The present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in different forms and should not be construed as limited to the embodiments set forth herein.

[26] FIG. 3 is a flowchart illustrating a 3DMC encoding process with texture coordinate quantization according to an embodiment of the present invention. As shown in FIG. 3, a bpt (bits per texture coordinate) value is determined (step 310). It is well known to those skilled in the art that the determination of the bpt value is a part of a typical 3DMC coding process and is not newly proposed by the present invention. In general, the bpt value may be determined by a user, regardless of the image size.

[27] An adaptive quantization step size is determined according to a method proposed by the present invention (step 320). The adaptive quantization step size is herein denoted by "delta". A delta value comprises "delta_u" used for quantization of a u-axis coordinate value and "delta_v" for quantization of a v-axis coordinate value. Hereinafter, the delta value is referred to as a value containing both "delta_u" and "delta_v".

[28] The bpt value is then compared to the number of bits to represent the delta value (step 330). If the bpt value is smaller than the number of bits to represent the delta value, the texture coordinates are quantized using the fixed quantization step size 2^{-bpt}, which is used in the conventional 3DMC process (step 340). On the other hand, if the number of bits to represent the delta value is smaller than or equal to the bpt value, the texture coordinates are quantized using the delta value (step 350). The process for obtaining the delta value according to an embodiment of the present invention will be described later with reference to FIG. 4.

[29] 3D mesh information including the quantized texture coordinates is encoded (step 360), and 3DMC packets with the delta information are generated and transmitted (step

370).

[30] FIG. 4 is a flowchart illustrating the process for calculating the adaptive quantization step size (i.e., delta) for texture coordinate quantization according to an embodiment of the present invention.

[31] First, it is determined whether the size information (image_size) of the texture image exists or not (step 410). When the size information of the texture image exists, the first adaptive quantization step size, delta1 (i.e., delta1_u, delta1_v) is calculated by the inverse of the image size (step 420). For example, when the image size is a*b, the delta1_u is 1/a, and delta1_v is 1/b. Alternatively, delta1_u and delta1_v may be 1/(a-1) and 1/(b-1), respectively.

[32] Next, the second adaptive quantization step size, delta2 (i.e., delta2_u, delta2_v) is estimated using the texture coordinate values (step 430). In one embodiment, delta2 may be determined to one of the mode value, the value of greatest common divisor (GCD), the median value, the average value, the minimum value, and the maximum value of difference values between the two neighboring texture coordinate values arranged in ascending order.

[33] It is determined whether the delta2 is a multiple of delta1 or not (step 440). When delta2 is a multiple of delta1, delta2 is determined as the adaptive quantization step size, and otherwise, delta1 is determined as the adaptive quantization step size.

[34] Meanwhile, when it is determined that the size information of the texture image does not exist at step 410, the adaptive quantization step size (delta) is determined using the texture coordinate values (step 470). The method of estimating the adaptive quantization step size (delta) at step 470 is the same as that of estimating delta2 at step 430. The adaptive quantization step size (delta) can be determined by various other manners .

[35] For example, when the texture image size is 800*400, since delta_u and delta_v, which are obtained according to an embodiment of the present invention, are close to divisors of 800 and 400 for u and v axes, the texture coordinate values can be re-constructed without any loss during the decoding process.

[36] In one embodiment, in order to calculate the optimum adaptive quantization step size, the filtering is performed on the real number texture coordinates within the original VRML file. Specifically, the real number texture coordinate value is multiplied by the texture image size, round up, down or off to obtain an integer value, and then divided by the texture image size, thereby obtaining the filtered real number texture coordinate values. Table 1 shows the results of filtering on the real number texture coordinate values when the texture image size is 800*400. And also, the filtering is performed in various other manners.

[37] Table 1

Original value (float)		Mapping value		filtered value (float)		Mapping value	
u	v	U	V	u	v	U	V
0.688477	0.643555	550	257	0.688360	0.644110	550	257
0.672852	0.643555	538	257	0.673342	0.644110	538	257
0.911133	0.604492	728	241	0.911139	0.604010	728	241
0.918945	0.612305	734	244	0.918648	0.611529	734	244
0.958008	0.530273	765	212	0.957447	0.531328	765	212

[38] FIG. 5 is a flowchart illustrating a 3DMC decoding process according to an embodiment of the present invention. 3DMC packets are received (step 510), and it is determined whether it contains the adaptive quantization step size (delta) information or not (step 520). The determination can be performed based on a flag in the 3DMC packet header indicating whether the delta information exists.

[39] When the delta information is not contained in the 3DMC packet, the texture coordinate values are inverse-quantized using the predetermined quantization step size, like the conventional 3DMC packet decoding process (step 530). On the other hand, when the delta information is contained in the 3DMC packet, the delta information is extracted from the 3DMC packet (step 540), and the texture coordinate values are inverse-quantized using the extracted delta information (step 550). The inverse-quantized texture coordinates are then decoded (step 560), and the remaining information within the 3DMC packets are also decoded (step 570). The 3D model may be reconstructed based on the 3D mesh information obtained at steps 560 and 570 (step 580).

[40] FIGS. 6a to 6d show the results of quantization of the texture coordinates according to the conventional 3DMC scheme and the present invention. The following 4 models available from a home page of MPEG-4-SNHC have been selected as test models.

[41] Table 2

Image	Image size
battery.jpg	600*400
earth.jpg	800*400
nefert131.jpg	512*512
Vase131.jpg	512*512
Vase212.jpg	mage512*512

[42] The first method ("Method 1") quantizes the texture coordinates using 2^{-bpt} as the

quantization step size according to the conventional 3DMC method. The second method ("Method 2") quantizes the texture coordinates using the first adaptive quantization step size, "delta1," (i.e., the inverse of the image size), proposed by the present invention. The third method ("Method 3") quantizes the texture coordinates using the second adaptive quantization step size ("delta2").

[43] FIG. 6a shows the encoding results of the original VRML (IFS) files of the test models using the above-described three methods. As shown, when Method 2 is used, all test models exhibit lossless compression with a bit reduction of about 10%. When Method 3 is used, the earth model exhibits lossless compression and bit reduction of about 40%.

[44] FIG. 6b shows the encoding results of the fixed VRML (IFS) files of the test models using the above-described three methods. As shown in FIG. 6b, Method 2 and Method 3 show lossless compression that there is no error/difference value between the original file and the reconstructed file with a compression rate being 10% to 40% higher.

[45] FIG. 6c shows the encoding results of the VRML (IFS) files of the test models according to the above-described three methods at the same bpt. As shown in FIG. 6c, the results of the present invention indicate better compression efficiency and a lower error rate.

[46] FIG. 6d shows the comparison of the encoding results between the VRML (IFS) files of the test models according to the Method 1 at the maximum bpt "16" and that according to the Method 2 or Method using the delta1 or delta2. As shown in FIG. 6d, in all test models, the present invention shows 40% to 65% higher compression rate and lower distortion than the conventional art. Also, it is understood that the conventional art cannot achieve lossless compression even though it uses the maximum bpt.

[47] FIG. 7a illustrates the rendering result after the encoding/decoding is performed according to the conventional 3DMC scheme; FIG. 7b illustrates the rendering result after the encoding/decoding is performed using the inverse of the image size as the delta value according to an embodiment of the present invention; and FIG. 7c illustrates the rendering result after the encoding/decoding is performed using the delta value calculated from the texture coordinates according to another embodiment of the present invention.

[48] FIG. 8 shows a structure of a 3DMC packet header with a flag "delta_flag" indicating whether there is adaptive quantization step size (delta) information in a 3DMC packet according to an embodiment of the present invention. When delta_flag is set to 1, it means that the 3DMC packet contains the delta value, i.e., delta_u and delta_v. Here, delta_u is a delta value for u-axis and delta_v is a delta value for v-axis.

[49] The present invention can be provided in the form of at least one computer readable program which is implemented in at least one product such as a floppy disk, a hard disk, a CD ROM, a flash memory card, a PROM, a RAM, a ROM, or a magnetic tape. The computer readable program can be implemented by a general programming language.

[50] As described above, the method of encoding/decoding the 3D mesh information for the effective texture mapping according to the present invention achieves lossless reconstruction of the texture coordinates by adaptively adjusting the quantization step size for quantizing the texture coordinates, thereby guaranteeing the accurate texture mapping.

[51] While the invention has been shown and described with reference to certain exemplary embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

Claims

[1] A method of encoding texture coordinates in 3D mesh information, the method comprising the steps of:
determining an adaptive quantization step size used for texture coordinate quantization;
quantizing the texture coordinates using the adaptive quantization step size; and
encoding the quantized texture coordinates.

[2] The method of claim 1, wherein the adaptive quantization step size is determined as the inverse of the texture image size.

[3] The method of claim 1, wherein the adaptive quantization step size is determined using the texture coordinates.

[4] The method of claim 3, wherein the adaptive quantization step size is determined to one of a mode value, a greatest common divisor, a median value, an average value, a minimum value, and a maximum value of difference values between the sorted texture coordinate values.

[5] The method of claim 1, wherein the step of determining the adaptive quantization step size comprises the sub-steps of:
checking whether the texture image size information exists or not;
determining the inverse of the texture image size as a first quantization step size when the texture image size information exists;
obtaining a second quantization step size using the texture coordinates;
checking whether the second quantization step size is a multiple of the first quantization step size;
determining the second quantization step size as the adaptive quantization step size when it is determined that the second quantization step size is a multiple of the first quantization step size; and
determining the first quantization step size as the adaptive quantization step size when it is determined that the second quantization step size is not a multiple of the first quantization step size.

[6] The method of claim 5, further comprising the step of obtaining the adaptive quantization step size using the texture coordinates when the texture image size information does not exist.

[7] The method of claim 1, further comprising the steps of:
setting a bpt (bits per texture coordinate) value of the texture coordinates;
comparing the bpt value with the number of bits to represent the adaptive quantization step size;
quantizing the texture coordinates using 2^{-bpt} when the bpt value is smaller than

the number of bits to represent the adaptive quantization step size; and
quantizing the texture coordinates using the adaptive quantization step size when the bpt value is greater than or equal to the number of bits to represent the adaptive quantization step size.

[8] The method of claim 1, further comprising the step of filtering the real number texture coordinates using the texture image size information.

[9] The method of claim 8, wherein the step of filtering the real number texture coordinates comprises the sub-steps of: for each real number texture coordinate value,
multiplying the real number texture coordinate value by the texture image size and round up, down or off the resultant value to obtain a corresponding integer texture coordinate; and
replacing the real number texture coordinate with a value obtained by dividing the corresponding integer texture coordinate by the texture image size.

[10] The method of claim 8, wherein the step of filtering the real number texture coordinate comprises the sub-steps of:
multiplying the real number texture coordinate value by (the texture image size minus 1) and round up, down or off the resultant value to obtain a corresponding integer texture coordinate; and
replacing the real number texture coordinate with a value obtained by dividing the corresponding integer texture coordinate by (the texture image size minus 1).

[11] A method of encoding 3D mesh information, the method comprising:
a first encoding step for encoding texture coordinates in the 3D mesh information according to any one of claims 1 to 10;
a second encoding step for encoding remaining information of the 3D mesh information; and
a step of producing 3D mesh coding (3DMC) packets which contain the 3D mesh information obtained by the first and second encoding steps and an adaptive quantization step size.

[12] A method of decoding texture coordinates in 3DMC packets, the method comprising the steps of:
extracting adaptive quantization step size information from the 3DMC packet;
inverse-quantizing the texture coordinates in the 3DMC packet using the extracted adaptive quantization step size; and
decoding the inverse-quantized texture coordinates.

[13] The method of claim 12, further comprising the step of determining whether the adaptive quantization step size information is contained in the 3DMC packet, wherein the texture coordinates are quantized using a predetermined quantization

step size when it is determined that the adaptive quantization step size information is not contained in the 3DMC packet.

[14] The method of claim 13, wherein the step of determining whether the adaptive quantization step size is contained in the 3DMC packet uses a flag in a header of the 3DMC packet, the flag indicating whether the adaptive quantization step size is used or not.

[15] A 3DMC decoding method, comprising the steps of:
(i) decoding texture coordinates in 3DMC packets according to any one of claims 12 to 14;
(ii) decoding the remaining information of the 3DMC packets; and
(iii) reconstructing a 3D model based on 3D mesh information generated from the decoding results in the steps (ii) and (iii).

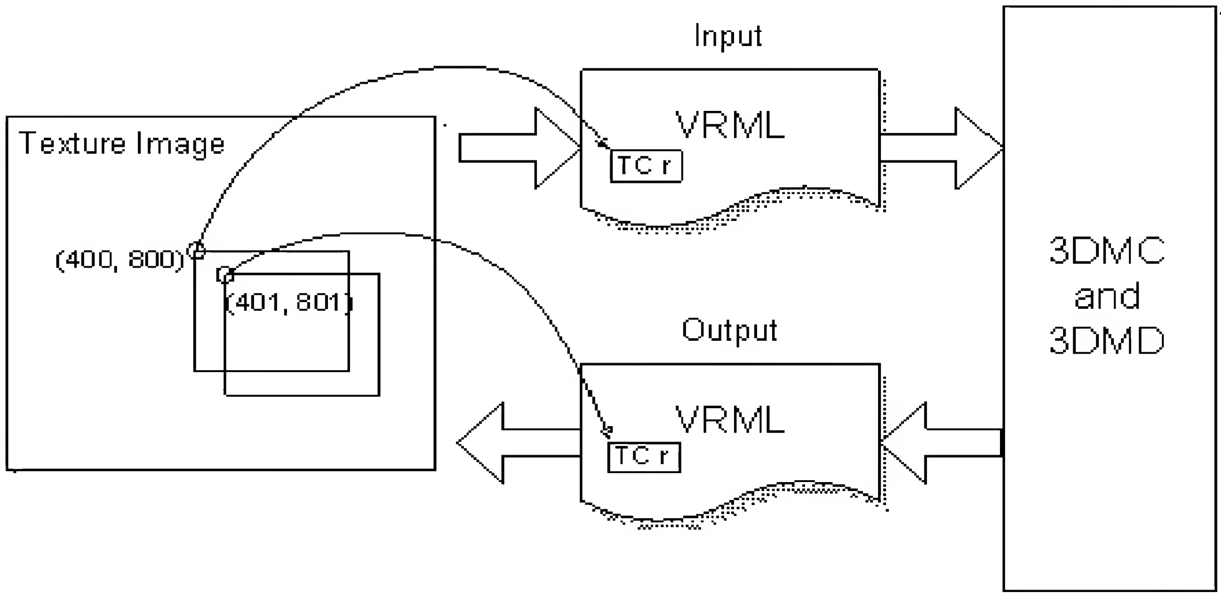
[16] A computer readable recording medium containing a computer program which performs the method of encoding texture coordinates in 3D mesh information according to any one of claims 1 to 10.

[17] A computer readable recording medium containing a computer program which performs the method of decoding texture coordinates in a 3DMC packet according to any one of claims 12 to 14.

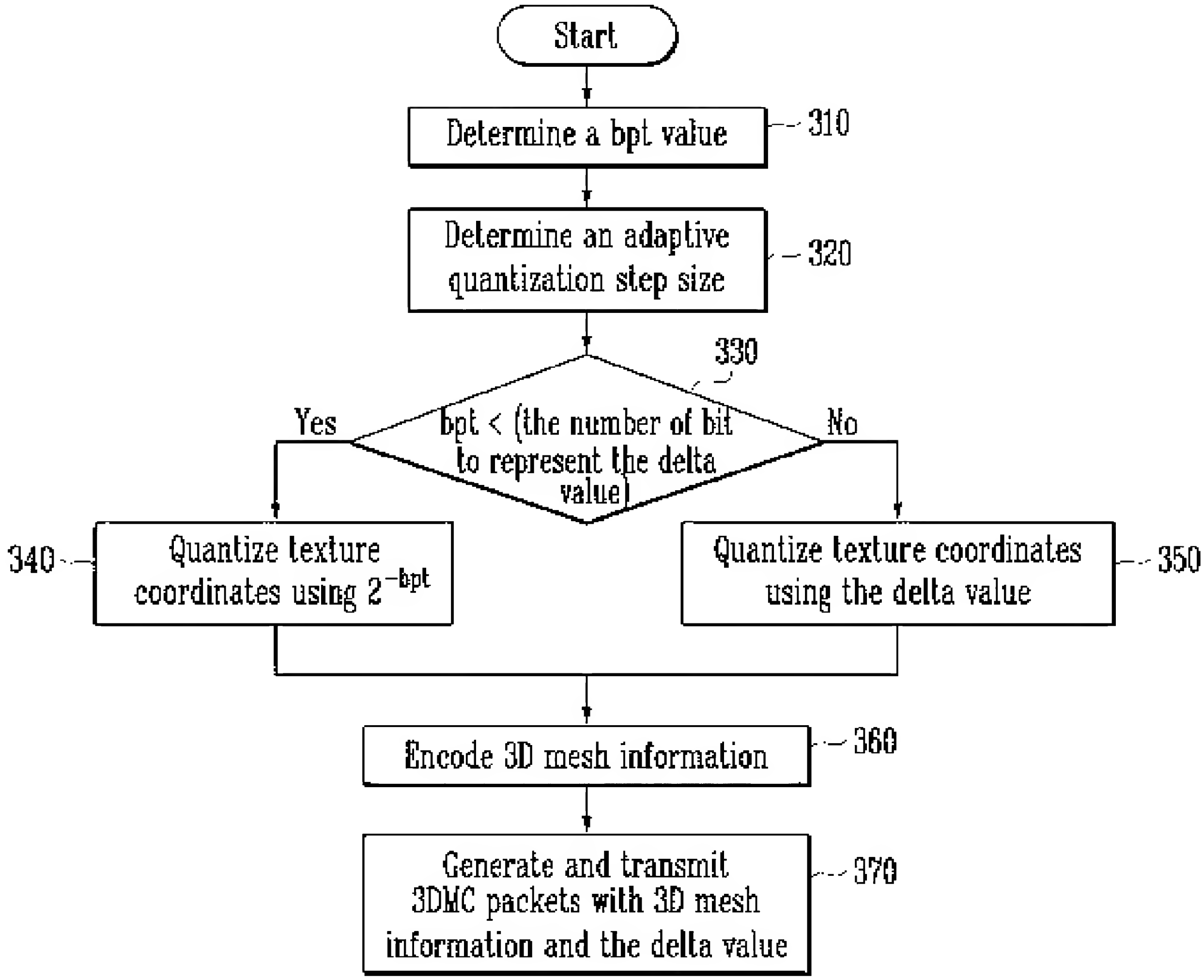
[Fig. 1]


IndexedFaceSet
3DMC
3DMC
BitStream

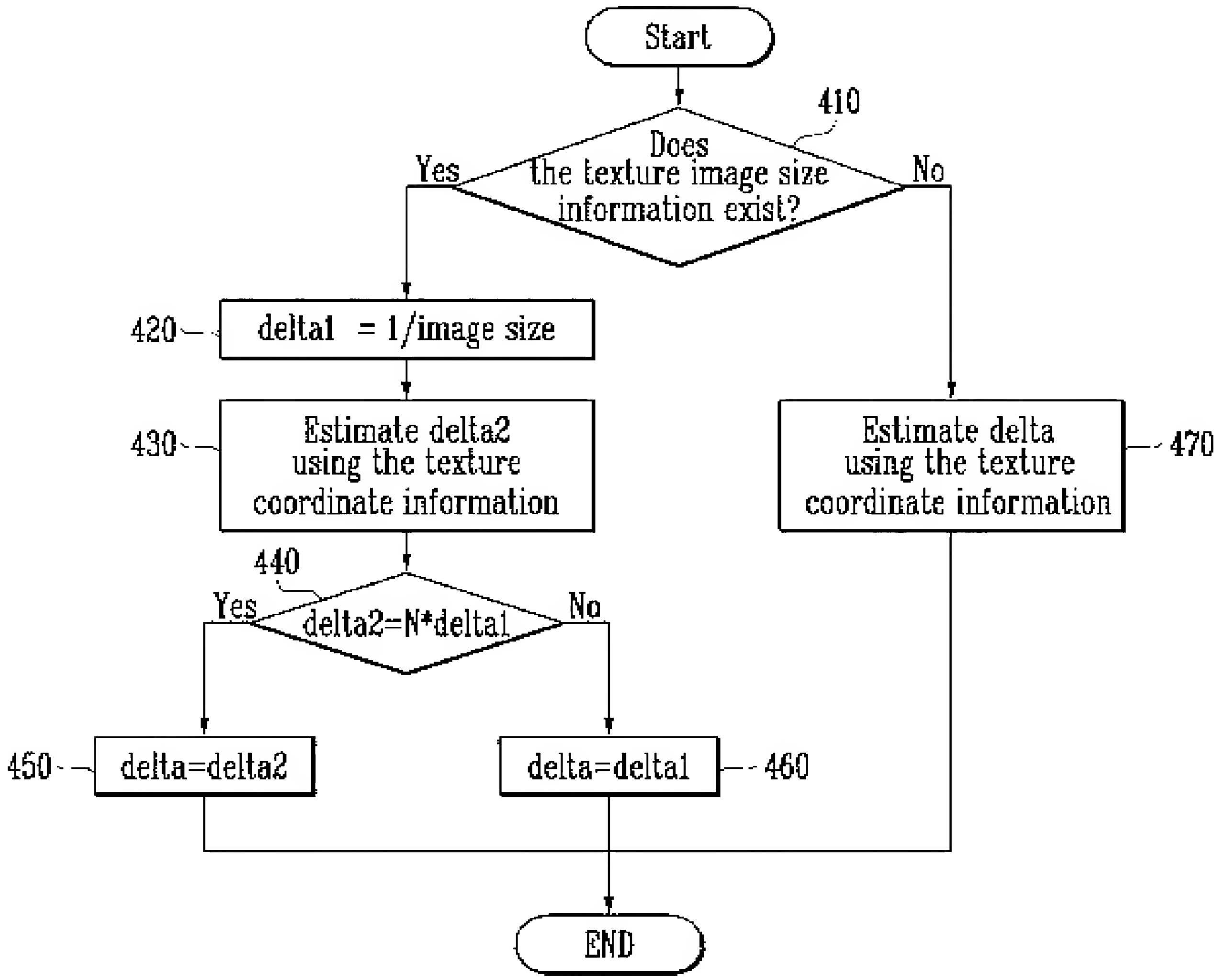
[Fig. 2]


Input
VRML
TCr
Texture Image
(400, 800)
(401, 801)
Output
VRML
TCr
3DMC
and
3DMD

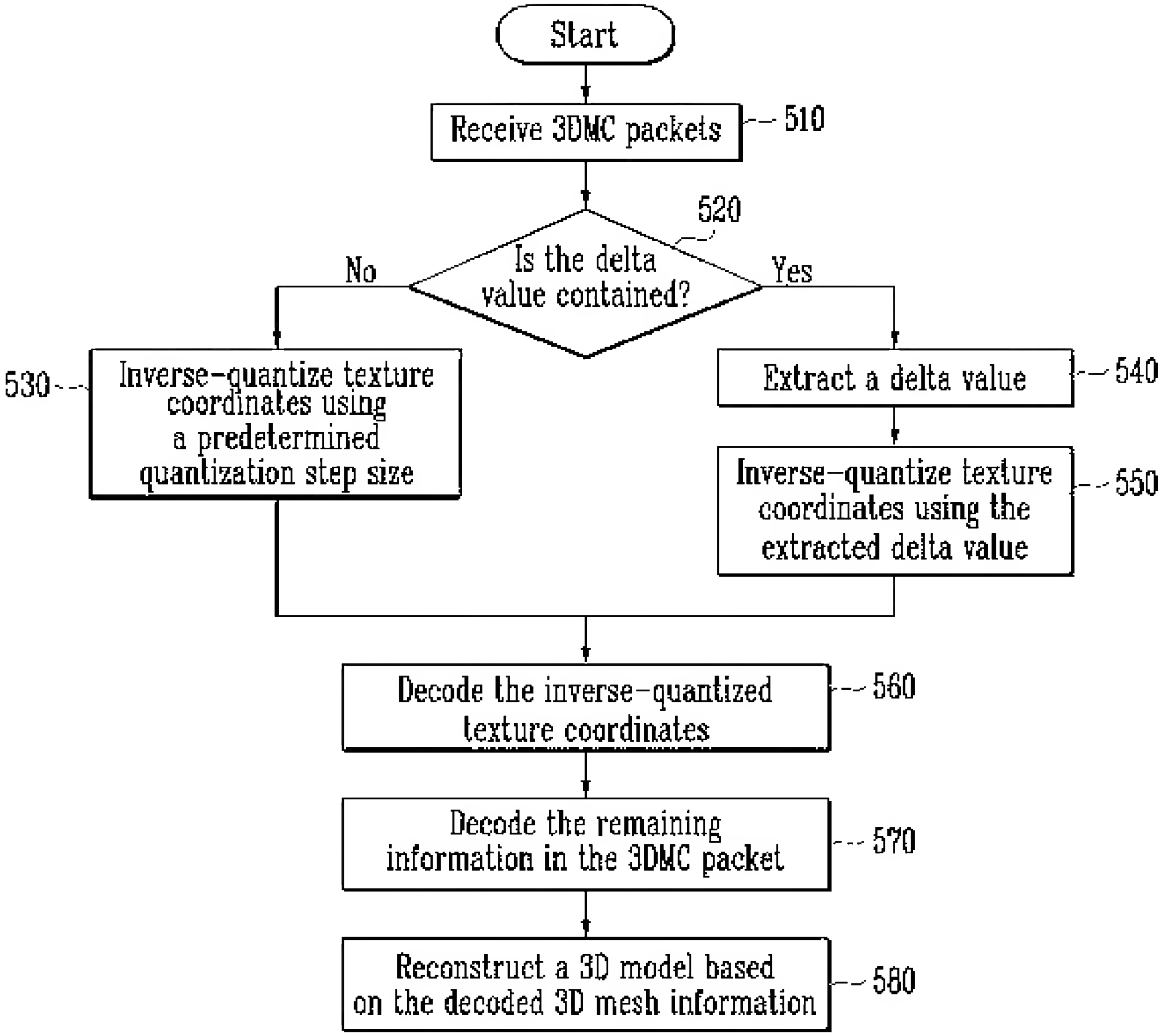
[Fig. 3]


Start
Determine a bpt value
310
Determine an adaptive quantization step size
320
330
bpt < (the number of bit to represent the delta value)
Yes
No
Quantize texture coordinates using 2^{-bpt}
340
Quantize texture coordinates using the delta value
350
Encode 3D mesh information
360
Generate and transmit 3DMC packets with 3D mesh information and the delta value
370

[Fig. 4]


Start
410
Does the texture image size information exist?
Yes
No
420
delta1 = 1/image size
430
Estimate delta2 using the texture coordinate information
470
Estimate delta using the texture coordinate information
440
delta2=N*delta1
Yes
No
450
delta=delta2
460
delta=delta1
END

[Fig. 5]


Start
Receive 3DMC packets
510
520
Is the delta value contained?
No
Yes
530
Inverse-quantize texture coordinates using a predetermined quantization step size
Extract a delta value
540
Inverse-quantize texture coordinates using the extracted delta value
550
Decode the inverse-quantized texture coordinates
560
Decode the remaining information in the 3DMC packet
570
Reconstruct a 3D model based on the decoded 3D mesh information
580

[Fig. 6]

(a)

	Method	MSE	Lossless count(%)	TC bits (%)	bpt
batteryD	1	0.8073	26.97	100.00	10
	2	0.0000	100.00	89.83	10
	3	0.0000	100.00	253.83	24
earth	1	0.9937	14.69	100.00	10
	2	0.0000	100.00	94.48	10
	3	0.0000	100.00	59.55	7
nefert131 b	1	0.8420	26.52	100.00	10
	2	0.0000	100.00	90.48	9
	3	0.8253	27.00	90.40	9
vase131b	1	0.8422	26.32	100.00	10
	2	0.0000	100.00	90.52	9
	3	0.8488	25.49	90.46	9
vase212b	1	0.8410	25.74	100.00	10
	2	0.0000	100.00	90.47	9
	3	0.8998	21.74	90.55	9

(b)

	Method	MSE	Lossless count(%)	TC bits (%)	bpt
batteryD	1	0.8002	24.82	100.00	10
	2	0.0000	100.00	93.68	10
	3	0.0000	100.00	93.68	10
earth	1	0.9422	12.24	100.00	10
	2	0.0000	100.00	94.43	10
	3	0.0000	100.00	59.52	7
nefert131 b	1	0.8426	25.56	100.00	10
	2	0.0000	100.00	90.48	9
	3	0.0000	100.00	90.48	9
vase131b	1	0.8441	25.65	100.00	10
	2	0.0000	100.00	90.52	9
	3	0.0000	100.00	90.52	9
vase212b	1	0.8264	27.72	100.00	10
	2	0.0000	100.00	90.47	9
	3	0.0000	100.00	90.47	9

(c)

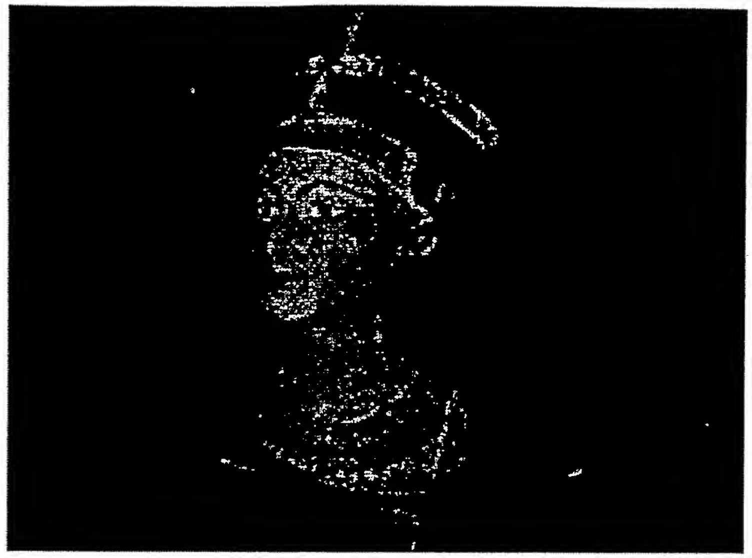
	Method	MSE	Lossless count(%)	TC bits (%)	bpt
batteryD	1	0.8073	26.97	100.00	10
	2	0.0000	100.00	89.83	10
earth	1	7.1108	0.00	100.00	7
	3	0.0000	100.00	82.86	7
nefert131 b	1	1.5963	10.96	100.00	9
	2	0.0000	100.00	100.11	9
vase131b	1	1.5665	9.17	100.00	9
	2	0.0000	100.00	100.06	9
vase212b	1	1.4167	0.07	100.00	9
	2	0.0000	100.00	100.02	9

(d)

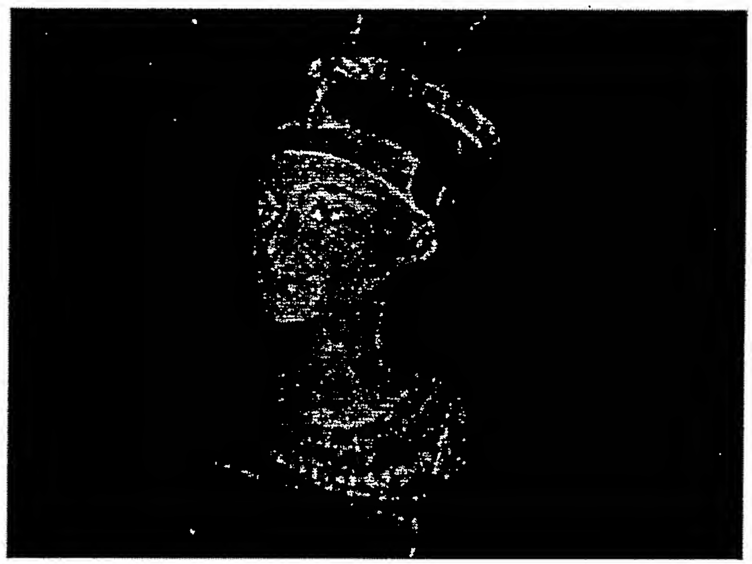
	Method	MSE	Lossless count(%)	TC bits(%)	bpt
batteryD	1	0.0126	98.76	100.00	16
	2	0.0000	100.00	50.45	10
earth	1	0.0303	96.98	100.00	16
	3	0.0000	100.00	34.88	7
nefert131 b	1	0.0211	97.90	100.00	16
	2	0.0000	100.00	57.37	9
vase131b	1	0.0220	97.80	100.00	16
	2	0.0000	100.00	57.58	9
vase212b	1	0.0388	96.15	100.00	16
	2	0.0000	100.00	57.49	9

[Fig. 7]

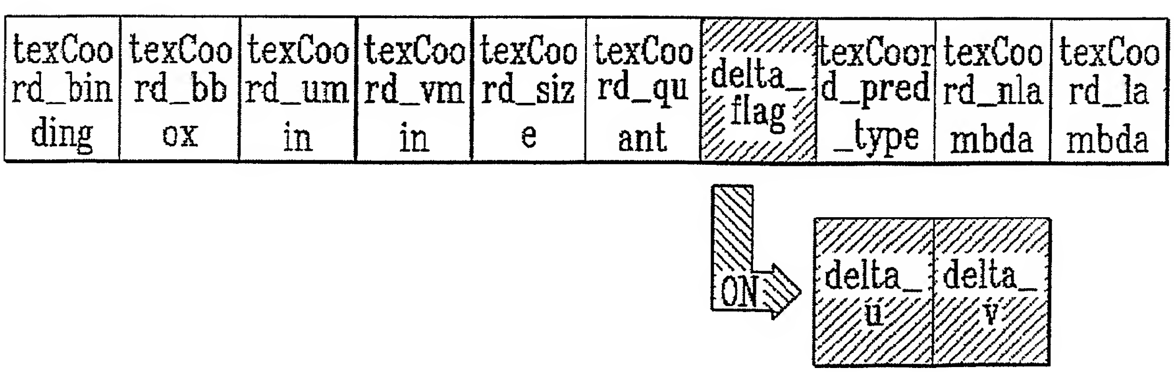
(a)



(b)



[Fig. 8]


texCoord_binding
texCoord_bbox
texCoord_umin
texCoord_vmin
texCoord_size
texCoord_quant
delta_flag
texCoord_pred_type
texCoord_nlambda
texCoord_lambda
ON
delta_u
delta_v

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2006/000151

A. CLASSIFICATION OF SUBJECT MATTER

G06T 9/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 8 G06T 9/00; 15/00; 15/50; 17/00;

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean Utility models and applictions for Utility models since 1975
Japanese Utility models and application for Utility models since 1975

Electronic data base consulted during the intertnational search (name of data base and, where practicable, search terms used)
WPI, ESPASNET, INSPECT, IEE/IEEE, PAJ, eKIPASS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,614,428 B1 (Jerome E. Lengyel) 2 Sep 2003 See the abstact	1-17
A	US 6,525,722 B1 (Michael F. Deering) 25 Feb 2003 See the abstract	1-17
A	US 5,870,097 A (John M. Snyder, etc.) 9 Feb 1999 See the abstract	1-17
A	US 6,573,890 B1 (Jerome E. Lengyel) 3 Jun 2003 See the abstract	1-17
A	US 6,593,925 B1 (Ziyad S. Hakura, etc.) 15 Jul 2003 See the abstract	1-17
A	KR 2003-0004943 A (Lee Sin-Jun) 15 Jan 2003 See the abstract	1-17

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 APRIL 2006 (24.04.2006)	**25 APRIL 2006 (25.04.2006)**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office 920 Dunsan-dong, Seo-gu, Daejeon 302-701, Republic of Korea	MA, Jung Youn
Facsimile No. 82-42-472-7140	Telephone No. 82-42-481-5679

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/KR2006/000151

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6,614,428 B1	2 Sep 2003	NONE	
US 6,525,722 B1	25 Feb 2003	AT 232624 E	15 Feb 2003
		AU 200057411 A1	2 Jan 2001
		DE 60001418 C0	20 Mar 2003
		EP 01194897 A1	10 Apr 2002
		WO 200077740 A1	21 Dec 2000
US 5,870,097 A	9 Feb 1999	NONE	
US 6,573,890 B1	3 Jun 2003	NONE	
US 6,593,925 B1	15 Jul 2003	NONE	
KR 2003-0004943 A	15 Jan 2003	CA 2372969 A1	28 Aug 2002
		CN 1384469 A	11 Dec 2002
		EP 1239680 A3	17 Mar 2004
		JP 14329217 A	15 Nov 2002
		RU 2226297 C2	27 Mar 2004
		US 20030142098 A1	31 Jul 2003